Kulpiszewski

METHODS TO CONTROL AND OPERATE A MESSAGE-SWITCHING NETWORK

G. Joachim Brandt and Georges J. Chretien

Société Internationale de Télécommunications Aeronautiques, Neuilly, FRANCE

In this paper we consider the principal methods to control and operate a store-and-forward message-switching network. Particular attention is given to the control of traffic, the routing, priority handling scheme and transmission errors, with special reference to the SITA network–this handling two types of traffic.

INTRODUCTION

The past few years have seen a substantial increase in the number of computer networks operating or being in the final stages of development.

The range of applications extends from a centralized computer system serving widespread terminals, such as airline reservation systems, communications networks interconnecting time-shared systems each sharing the resources of the other (such as the ARPA network), to public data networks such as the proposed Experimental Package Switching Service (EPSS) of the British Post Office. The methods to control such computer networks are numerous and it seems that the operating principles and design rules are still a long way from a consolidated and standardized approach.

Specific requirements, such as the type and geographical distribution of terminals to be interconnected as traffic sources and destinations, message characteristics, the service requirements in terms of transit times, plus reliability, govern the control and operating methods to such an extent that it is difficult to establish any general design patterns.

The problems are less of course if a network can be designed specifically to suit a definite prescribed need, but often in an operating system the requirements evolve after the system has been designed and implemented and the network has to be enlarged and modified to cope, instead of being designed precisely for the increased and often more stringent requirements.

The extensions to the network and the increasing traffic volume it has to handle may bring in their wake modifications to the control and operating procedures, which modifications have to be effected taking into account the existing network system, both its hardware and software.

This paper attempts to give guidance in deciding on a number of options which are available for the control and operation of a medium-speed message-switching computer

---

Presented at the Symposium on Computer-Communications Networks and Teletraffic
Polytechnic Institute of Brooklyn, April 4–6, 1972.

network. In order to have a representative example to study, we have taken for the purposes of our paper the data switching network which is operated by SITA (Société Internationale de Télécommunications Aéronautiques) to provide communications facilities for their member airlines.

In preparing these notes we have drawn largely on experience gained in operating this network from the development stages to its present widely extended configuration.

The user requirements and the traffic characteristics which determine the design of a communications network are discussed, particular reference being made to the SITA network, its structure and transmission procedures to guard against errors.

Topological and economic considerations lead from the traffic characteristics to a geographical allotment of switching and transmission capacities, that is, to a basic network structure, and some of the considerations of general interest which led to the structure of the SITA network are described in the following.

The next step in the network design is to evaluate whether, by means of proper routing and control strategies, the required transit times can be guaranteed, otherwise the network structure will iteratively be modified. In the designing of traffic control functions particular consideration has to be given to the avoiding of center overloads which can be caused by limited processing and storage capacities.

The paper turns then to discussion of the appropriate assignment of priorities to the various types of messages, and describes in the concluding chapter the calculation model used to determine the transit times through the network.

THE SITA DATA-SWITCHING NETWORK

The SITA network has grown from its initial operation of a number of shared low speed telegraph circuits for the exchange of telegrams into the present world-wide data network serving more than 150 airlines. Today the majority of the 110 switching centers are still manually operated, but the computerized subnetwork, operating since 1968, will by the end of this year comprise more than 25 computer centers, consisting of 11 high level switching centers (Philips DS-714 Mark II or Univac 418-II or -III) and 23 Satellite Processors (SPs) (Raytheon 706 or Thomson Houston TH-AC 4020). Each high level Center is equipped with mass storage to provide the required security for Type-B messages. The configuration of the network is shown in Figure 1. High level switching centers are indicated by circles and Satellite Processors by triangles (within the circles where these are operated on site).

The dashed lines indicate high level Centers not yet in operation. All interconnecting lines are voice grade circuits, currently operated at 2400 bps, the intention being that wherever possible the speed will be upgraded to 4800 bps during the course of 1972. The major part of the traffic handled by the SITA network is still telegraph or so-called Type-*B* traffic, but the conversational Type-*A* traffic is becoming progressively more important. In order to reduce their communications costs, an increasing number of airlines are considering the use of the SITA data network as real-time interface between their internationally distributed CRTs and the central Reservations Processor (RP). Thus the expected dominant share of Type-*A* traffic determines the further development of the network and its control procedures.

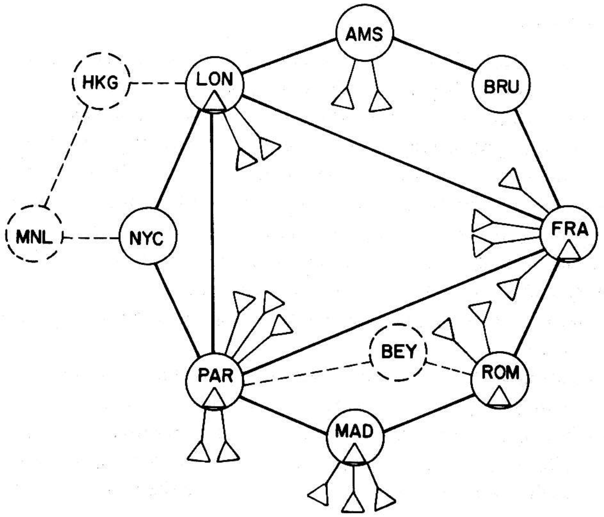

Fig. 1. The SITA data-switching network.

Traffic Characteristics and Service Requirements

For conversational Type-*A* traffic the network has to provide transit times of less than 1 sec, giving to the operator the impression of using a CRT which though remote is directly connected to the parent processor. One absolute requirement for this communications interface is a high degree of availability.

Mean Type-*A* message length is about 300 bits for the enquiry message (CRT to RP) and 500 bits for the response message (RP to CRT). The response time requirements take precedence over the requirements for safe transmission, in so far that in the (extremely remote) event that a message happens to be involved in a switching center failure, it is preferable that the message be discarded, rather than to provide additional security to ensure delivery, even though delayed, as this late arrival could cause considerable confusion having in mind the fact that if a response is missing the operator will regenerate his enquiry.

For Type-*B* traffic the requirements are quite different. As soon as the message is accepted from the outstation, the network takes over responsibility for complete protection against loss. Additionally, after being delivered, the messages must be kept in mass memory to allow for retrieval and individual message accounting.

If Type-*B* messages cannot be delivered because of failure of an outstation, these would wait on drum until they can be delivered, but Type-*A* messages in such a case would be discarded.

Type-*B* messages are in general one-way (or the answer takes so long that they can be considered as one-direction messages) and have transit time requirements in the order of minutes. The address analysis (seven-letter low level address, to be identified from a comprehensive list), added to the fact that a high percentage of the messages are multi-address and have to be delivered separately, plus verification of the originating address, format checks, and eventual code translations, results in processing times for these Type-*B* messages in the order of a hundred milliseconds, as opposed to Type-*A* processing times which are in the order of 10 msec.

Network Structure

The SITA data network (Fig. 1) is a distributed-communications network operating in a store-and-forward mode. One of the inherent advantages of such a system design is that line and center outages cause only limited harm. As far as line outages are concerned, which, in the case of a centralized network can result in isolation of a complete area, these in a distributed-communications network are catered for by an alternative routing procedure, every nodal point being connected to at least two other nodal points. As far as center outages are concerned (a very rare event and the probability of which is further reduced by the use of dual systems), in a centralized communications network failure of the central switch causes complete shut-down of the entire system, while in distributed communications the only terminals affected by the outage are those controlled by the center which is down.

The concentration of terminals controlled by one switching center, advantageous from the point of view of capacity sharing, has also to be considered having in mind the benefits of distributed control. Figure 2 shows a possible configuration of a high level switching center, illustrating the flexibility required as far as concerns connectibility of terminals, extensibility and the possibility to reconfigure the switching centers.

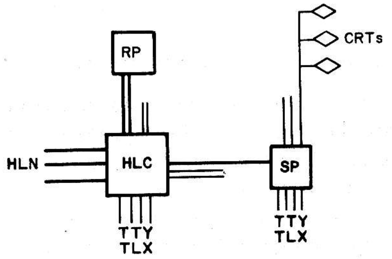

Fig. 2. Configuration of a SITA high level center.

Applications processors are connected to only one switching center, perhaps using a multicircuit link following different routes as a safeguard against link outages. While this, in case of failure of the switching center, isolates the applications processor, this risk is accepted because to connect these to two switching centers would involve them in the overall control and routing procedures of the network.

The SITA decision to opt for a store-and-forward-type network was quite straightforward, this offering the best combination of minimum communications costs with maximum availability of the system, while at the same time fulfilling transit time requirements.

Another inherent advantage of this distributed communications principle—and an essential need in the case of a continually expanding (geographically and in terms of traffic volumes) multi-user network—is that the network can be easily extended without necessarily affecting or causing modification or enlargement of the parts already existing.

Procedures to Guard against Transmission Errors and Loss of Messages

Protection has to be provided to cover two different areas of error source:

(1) Transmission errors
(2) Loss of messages due to computer failure.

In store-and-forward systems, messages are routed through the network step by step from processor to processor. If messages are handled and stored in the switching centers in core, without recourse to the drum, this is termed core switching. In the processor to processor transmission, protection against transmission errors is provided for all types of messages. In the SITA system this is realised by a free-wheeling link control procedure, the basic features of which are:

(a) Two types of information messages are accommodated, Type *A* having nonpreemptive priority over Type *B*. Control messages have nonpreemptive priority over Types *A* and *B*
(b) Messages exceeding a pre-established length are subdivided into transmission blocks, for two reasons:
    (i) To allow a higher priority message to be sent after transmission of a block instead of having to wait for the completion of an entire message, thus avoiding excessive delays for higher priority messages
    (ii) To optimize the number of bits checked by one control sequence with respect to the transmission error rate
    The pre-established maximum block length results from an optimization between criteria (i) and (ii) and the distribution of message length
(c) The blocks which are forwarded are stored in the sending center until they have been acknowledged by a special control message
(d) In the block envelope a numbering scheme and a control character are provided. The numbering scheme gives an identification number so that the block is uniquely registered. The control character allows the receiving center to detect transmission errors
(e) In case a transmission error is detected, the incorrect block is discarded and the sending center, after having received a negative acknowledgement link control block, or after a time-out without having received a positive acknowledgement, will repeat the incorrect and any subsequent blocks
(f) When a center has accepted and acknowledged a block it is responsible for that block until it in turn receives an acknowledgement from tne next center on the transmission route, in which event it can release the block.

The interconnection of functional processors to the SITA network is effected by means of standard link control procedures developed by ATA/IATA. For Type-*A* messages the core switched transfer of messages from processor to processor provides a sufficiently high measure of security. In the case of Type-*B* messages additional safeguards have to be provided against the loss of messages during computer outage. This is achieved by a drum to drum transfer superimposed on the core to core transfer. As soon as the Type-*B* message is received by the communications system entry center it is stored safely on drum. The message is released from the entry drum only upon receipt of a special control message which indicates to the entry center that the message is now safely stored on drum in the exit center. In transit centers, Type-*B* messages are handled in the same way as Type-*A* messages, that is are core switched, but with lower priority. Due to the free-wheeling transmission procedure the transit centers have no means to control the influx of transit Type-*B* messages and during peak hours this contributes to any problem caused by lack of buffer space. An alternative method is that immediately after a transit Type-*B* message is received it is transferred to mass memory, where it waits until it can be forwarded. This frees core space for Type-*A* messages.

Controlled vs Free-Wheeling Mode of Operation

There are in principle two different modes of operating communications networks; *in principle* because in practice a network is generally operated by a combination of these two modes. In a free-wheeling environment a terminal or a switching center is able to generate or forward a message at any time and the receiving side has to be ready at all times to accept these messages, i. e., the receiving side is able to cope with every traffic peak without rejecting any incoming traffic. The alternative mode of operating a communications facility is to completely control the number of incoming messages. A typical example is a centralized reservations network, in which the widely spread terminals are polled directly by the central processor. Terminals which have messages to send have to wait until they are solicited, that is until the receiving center signals that it is ready to accept traffic. In the event that the processor is nearing overload situation, in terms of running short of storage or processing capacity, it is able to reduce the polling rate, thereby slowing down the receipt of messages, or it can stop polling completely. By delegating to the outstations the task of storing messages which cannot be handled immediately by the communications system, the storage and processing requirements in the switching centers are quite notably reduced and the capacity of the centers need not be as high as to cater for the highest traffic peak that might occur.

The main disadvantage of a controlled mode of operation is increased processing and line load due to the enlarged overhead per message. In a distributed-communications network control procedures between the switching centers would increase to quite a large extent the internal transit times, at least in a medium speed network. Whereas in a centralized network the central processor can control the load on the entire network, such overall load control cannot be achieved in a distributed-communications network where each switching center is only able to control and regulate the inflow of traffic from its connected terminals, according to its own capacity or condition, not knowing the load situation in other centers and whether or not those other centers are able to accept further traffic. Such a control could be achieved by regularly sending to all centers information concerning center loads but this would call for each center being kept completely up to date on the status and load of all other centers. Depending on the destination of an incoming message, the switching center would have to decide (a lengthy and impractical process) whether or not the network as a whole is able to handle this message. This is obviously a prohibitive undertaking from all points of view.

One possibility is to limit the total number of messages in the communications network at one time, for instance by means comparable to the logical path concept in the ARPA network. For a communications system interfacing reservation systems and other terminals, a direct control and regulation of the traffic flow is not an appropriate method, at least not if the transmission lines are operated as medium speed links. SITA therefore adopted an alternative means of controlling the traffic flow. This traffic regulation principle can be described as free-wheeling for Type-*A* and controlled mode of operation for Type-*B*.

The processing and transmission capacities of the SITA network have been assigned according to the total load of Type-*A* plus Type-*B* traffic, which capacity can be temporarily reserved for Type-*A* in order to cope with Type-*A* peak traffic situations. A center will have to accept incoming Type-*B* messages even in overload situations. These messages will be immediately transferred to drum, where they wait for processing, which can be started as soon as the overload situation has been removed. The same argument holds

for the availability of buffer storage capacity, which is assigned according to the cumulative load of Types-*A* and *B* traffic, but which will, if necessary, be reserved for Type-*A* messages. Due to the fact that we can temporarily reserve the network resources for Type-*A* traffic we can cope with most of the peak Type-*A* traffic situations without the centers running into overload condition.

Routing of Messages

The network we consider is for civilian use, which means it is not subject to constraints as strict as those imposed on military networks. The size of the network in our example is less important and furthermore its vulnerability has nothing in common with that of military networks. However, the basic requirements of the two are the same; each node of the network should be informed, directly or indirectly, about the overall network configuration changes and the status of all the other centers in the network, and provision should be made for link redundancy and node reliability to ensure that the availability of the network is very high. There is a further requirement for our network; changes in the status of a link or center should be reported as quickly as possible (mainly for Type-*A*) to allow rerouting of messages by the fastest route and to avoid regeneration of Type-*A* traffic (if the answer takes too long to arrive), as this would increase the traffic load.

Many routing schemes have been proposed [1-4] and these can be classified in two categories:

(a) Adaptive routing, based on the momentary load of the network
(b) Alternative routing, based on the average input traffic matrix of the network.

Stochastic adaptive techniques are ruled out since they cannot be adapted rapidly enough to the configuration changes. Irrespective of the type of routing used, we assume that between two updatings of the routing tables in a node only one possible route will be chosen to reach one center:

(1) Adaptive routing based on the momentary load of the network allows very efficient use of the facilities and there are two ways of realizing this routing:
    (a) The information related to one node (changes in the status of the links or the center, queues on the outgoing links, load of the center) is sent by this node to all the nodes of the network, independently and whenever necessary. Each node will then compute and update its routing table
    (b) Every *T* seconds each node sends to its adjacent centers information (the delays) concerning its own outgoing circuits and will in turn receive routing information from adjacent centers. Then, based on the already available information, allied to this new incoming information, it will update its routing table. This method is sometimes referred to as the distributed solution for adaptive routing
(2) Alternative routing, based on the average input traffic matrix, is a much simpler approach to the routing problem. With this method the network is never fully adapted to the actual traffic, or, in other words, the transit time of a particular message will not be optimised with respect to the actual traffic. Here each center will just send the information it has concerning the status of its adjacent links

(3) The method described in (1a) is a straightforward method but requires a large-scale routing program and storage, is time-consuming, and can only be implemented in very small networks, provided the updating of information is not too frequent, otherwise the time devoted to computing the new tables may lower the switching performance of the centers. The method described in (1b) [2 and 3] allows each center to compute its own routing in a much easier way than by the previous method. Routing will be based on the almost-actual load of the data links. This method may be termed as distributed and synchronous, i.e., updating of the routing table in each center is based only on the information sent regularly by the adjacent centers at given time intervals. This implies the choice of an *updating period* and that all the centers are synchronized. The choice of the updating period is critical. If, for instance, a link goes down, a center three links distant from the one impaired will get the information after three time intervals. If the updating period is long compared to the transmission time this might cause messages to bounce back and forth between two centers during, for instance, one updating period.

(4) The alternative routing will very rapidly provide an alternative route and will adapt immediately to changes in the network. A practical way to compute the routing is the minimum link traffic method

(5) One proposed solution is the following: The probability of getting more than 3 independent links down at the same time (that is not just declared down due to outage of the center to which they are connected) is very remote. Therefore our main concern is with configuration changes when one or two links change their status, or when a center is isolated. The routing for this different situation is then computed off-line, based on the minimum link traffic method, and the different tables are stored in the computers. If more than two independent links, or more than one node is down, the resulting situation will be considered an emergency in which the efficient optimum routing is not too important compared with the necessity to forward messages on their way to their destination. For that situation another table, based on the shortest path, will be established and stored in the computer. This cannot be considered as an approach to the problem of routing, but rather as a solution dedicated to a specific small network (in the present network 8 nodes) and this is the type of solution which is now implemented in the SITA network. We are now considering the possibility of implementing an adaptive routing scheme in this network. However, if this is done we would superimpose on this routing, control messages which would be sent to all the centers immediately a link or center is down. Such a scheme is of course part of our existing network control and it could be described in the following way: Any change in the link status will be recognized by the two adjacent centers connected to the particular link, and they will generate information messages which may be sent specifically to each center, or (having no specific destination center) could be relayed over the network by each center. Any center receiving such an information message will update its routing table. In the case of center failure, the procedure is identical to that described above: All the links connected to the failed center will be reported down

(6) Whatever the routing scheme used, we have also to consider the possibility that

such networks will extend rapidly, and in our particular example we could expect extension of the network to cover different locations which are geographically far apart. One possible way then to enlarge the complete communications system would be to integrate these centers with the already existing network, but this would result in a very large network, the routing would become more difficult while at the same time we could expect the exchange of traffic to be mainly locally distributed. From this it can be seen that a more reasonable approach would be to consider the different areas as subnetworks, these being interconnected but not integrated. To route the traffic from one subnetwork to another we would use an alternative routing scheme based only on the average traffic and shortest path between these subnetworks. This could be termed as a method with compression of the routing information, i.e., each subnetwork has only the information corresponding to its centers and to the connecting links with different subnetworks.

PRIORITY HANDLING SCHEME

In describing the priority handling scheme we confine ourselves to the order of priority in which outgoing messages are sent. In this section we first describe the priority structure applied to the various types of messages. We then discuss the advantages to be gained by introducing preemptive priority as opposed to nonpreemptive priority. Without going into too much detail we differentiate between three priority classes: Highest priority are link and network control messages; second are Type-*A* messages; and third priority are Type-B messages. The present link control procedures handle the traffic based on a nonpreemptive priority scheme, which means that higher priority blocks cannot interrupt those of lower priority but have to wait until transmission of the lower priority blocks is completed. The maximum block length (on a 2400 bps line) corresponds to some 800 msec transmission time and thus the delays imposed are quite considerable.

What gains in terms of transit times could be achieved by introducing preemptive priority? Depending on the total load on the outgoing line, the message length distribution of the different priority classes and the mix of Type-*A* and Type-*B* messages, the preemptive discipline quite effectively reduces the waiting time for link control and Type-*A* messages, but here we are more interested in the queuing time for Type-*A* messages.

In order to take into account the load and delays caused by high priority control messages, it has been assumed that the number of control messages on the outgoing line equals the number of information messages, thereby indirectly assuming that the corresponding incoming line has the same load. For an example we have chosen the message characteristics of the SITA network, which are:

Line speed:	2400 bps,
Length of control messages:	40 bits,
Mean length of Type-*A* messages:	400 bits,
Mean length of Type-*B* messages:	1600 bits.

This results in the following table:

Arrival rate [messages per second]

Link control messages	$\lambda_1 = \lambda_2 + \lambda_3$,
Type-*A* messages	λ_2 ,
Type-*B* messages	λ_3 .

Service Time [sec]

	First order moment	*Second order moment*
Link control messages	$\beta_1^{(1)} = \frac{1}{60}$	$\beta_1^{(2)} = [\beta_1^{(1)}]^2$
Type-*A* messages	$\beta_2^{(1)} = \frac{1}{6}$	$\beta_2^{(2)} = 1.7\,[\beta_2^{(1)}]^2$
Type-*B* messages	$\beta_3^{(1)} = \frac{2}{3}$	$\beta_3^{(2)} = 1.4\,[\beta_3^{(1)}]^2$

Load

$$\rho = \sum_{i=1}^{3} \lambda_i \beta_i^{(1)} .$$

The corresponding formulae for the queuing time, that is waiting time plus service time for preemptive and nonpreemptive priorities, are as follows:

For Type A

Queuing time [sec]

Nonpreemptive

$$q_2 = \frac{1}{2}\,\frac{\sum_{i=1}^{3} \lambda_i \beta_i^{(2)}}{[1 - \lambda_1\beta_1^{(1)}]\,[1 - \lambda_1\beta_1^{(1)} - \lambda_2\beta_2^{(1)}]} + \beta_2^{(1)} .$$

Preemptive

$$q_2 = \frac{1}{2}\,\frac{\sum_{i=1}^{2} \lambda_i \beta_i^{(2)}}{[1 - \lambda_1\beta_1^{(1)}]\,[1 - \lambda_1\beta_1^{(1)} - \lambda_2\beta_2^{(1)}]} + \frac{\beta_2^{(1)}}{1 - \lambda_1\beta_1^{(1)}} .$$

In Figure 3 the different queuing times for Type-*A* messages are represented for various mixes and loads (ρ).

Curves 1 and 2 represent the queuing time with nonpreemptive discipline, while curves 3 and 4 represent the queuing time with preemptive discipline for the same mixes. Curve 5 + 6 represents the queuing time when there are no Type-*B* messages ($\lambda_3 = 0$) for both disciplines.

For a given load, (we assume $\rho < 0.7$) as the proportion of Type-*A* traffic increases, the average queuing time for this traffic will increase in the preemptive priority scheme

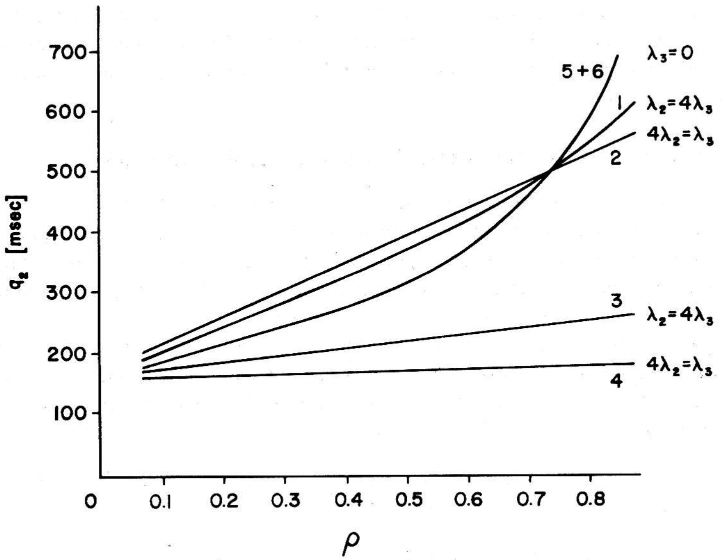

Fig. 3. Queuing time for Type-*A* messages under preemptive and nonpreemptive disciplines.

whereas in the nonpreemptive priority scheme it will decrease. When there are not many Type-*A* messages the preemptive priority allows reduced queuing time, for example reduced by more than 50% (see curves 1 and 3).

A great benefit can be obtained by using preemptive priority because the ratio of the second order moment of Type-*A* and Type-*B* messages is very low. Apart from the fact that introduction of preemptive priority considerably reduces the transit times of Type-*A* messages through the network, it brings about at the same time a reduction in buffer storage requirements. This reduction is due to the specific handling of Type-*A* and Type-*B* messages, as the delayed Type-*B* messages can be kept on drum, contrary to Type-*A* which have to be maintained in core.

We have discussed the introduction of preemptive priority of Type-*A* over Type-*B* messages. Would there be any point in giving control messages preemptive priority over information messages? In fact this priority handling would not reduce the transit times of the information messages because these are forwarded as soon as they arrive in switching center and do not have to wait until the ACK is sent via the link on which they arrived. The shortened waiting time for the acknowledgement lessens the storage requirements for those messages already transmitted and which are awaiting acknowledgement before they can be released.

An additional considerable advantage of giving preemptive priority to control messages is the short transit times, in the order to 10 msec even on a medium speed network. Network and routing information can be forwarded very rapidly, about 15 msec, to its destination. This is comparable with the transit times for control messages in high speed networks with nonpreemptive priority. The benefits of introducing a preemptive priority largely depend on transmission speed and the transit time required. In our example of a medium speed network these gains can be notable but in a high speed network with transmission speeds of 50 kbits, the relative gains are obviously much smaller. In other words, generally speaking, the increase in transmission speed, as long as the costs of such increase in terms of capacity or traffic volume can be justified, is preferable to the introduction of preemptive priority as at the same time it reduces the waiting and transmission times.

Calculation Model for the Transit Time through the Network

The total transit time through the network can be decomposed into three parts: transmission time, waiting time in front of the outgoing lines, and handling time in the switching center, which includes processing time and the time spent waiting for processing. The relative percentages give indication as to which of the three parts, if reduced, would result in the greatest benefit to the transit times.

We give an example for the transit time through the SITA network. The calculated transit times are based on a network model using the assumption of independence between the nodes in a store and forward system, and we have the average transit time for Type-*A* messages in the network

$$T = \frac{1}{\gamma} \sum_{j} \alpha_{2j} \, [q_{2j} + t_{p_0} + t_p] \; ,$$

in which γ is the flow of Type-*A* traffic entering the network per second; q_{2j} : mean

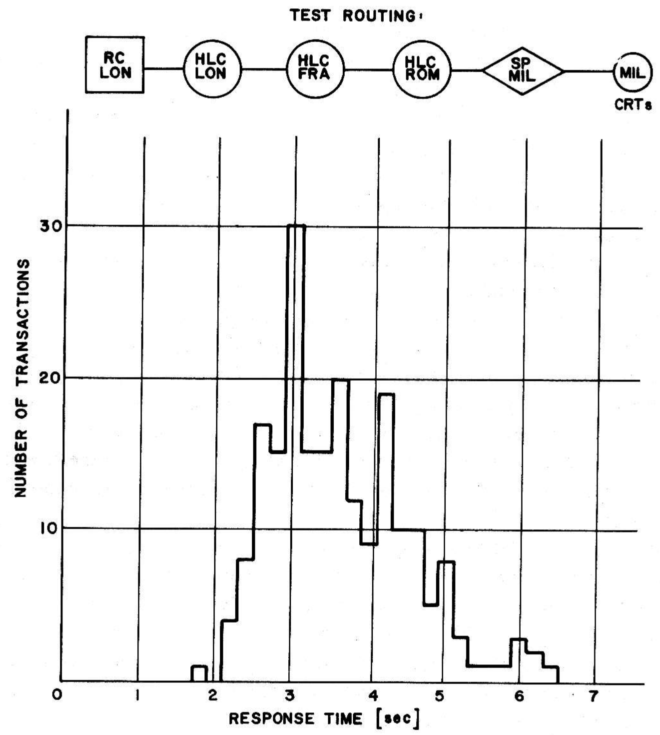

Fig. 4. Response time distribution for Type-*A* messages.

queuing time for nonpreemptive case on link Q_j (see Section Priority Handling Scheme); t_{p_0} : propagation time; t_p : average processing time for Type-*A* message in a center; and α_{2j} : average Type-*A* traffic on link *j*.

As has been demonstrated in papers, for instance from Kleinrock, this hypothesis is justified and gives reliable results, in particular when the number of traffic sources, that is incoming lines, is high and results in a good traffic mix. In fact this has been confirmed by measurements taken in live traffic conditions between a reservation computer in London (RC LON) and CRTs (MIL) connected to one of our Satellite Processors in Milan (SP MIL). The Type-*A* messages were generated in Milan (CRTs MIL) and sent to (RC LON) via the high level centers of ROME (HLC ROM), FRANKFURT (HLC FRA), LONDON (HLC LON). After some processing the RC LON sent back an answer via the same path to CRTs MIL. The response times measured, i.e., time elapsed between the sending of the first character of the query and the receipt of the first character of the answer, are shown in Figure 4 for one typical sample of Type-*A* messages. The mean response time observed was 3.8 sec. The calculated response time, based on the above-mentioned hypothesis, was 3.5 sec.

The planned upspeeding to 4800 bps of all the voice grade channels used on the SITA network will bring about a further considerable reduction in the response time, which it is estimated will be about 2.6 sec, that is a reduction of 25%.

CONCLUSIONS

In the preceding paragraphs we have described different methods to handle and control traffic of two message types in a communications computer network. To improve the operation of the network used in our example we will now consider the following possibilities:

(a) To include complete control of the incoming traffic sources
(b) Eventually an adaptive routing scheme, although we would still maintain the present scheme whereby all centers are instantly informed about any status changes occurring in the network
(c) Priority handling scheme: Although preemptive priority gives important gains in transit times, as we expect a very high increase of Type-*A* traffic on the network, this benefit would become less important with respect to nonpreemptive priority
(d) • Fully transparent procedure that will improve the line efficiency while at the same time allowing the SITA network to receive any type of traffic without any restriction as to bit patterns
    • Storage of Type-*B* messages on mass storage units in transit centers
(e) Transit times could easily be improved, with negligible increase in cost, by upgrading the speed of our circuits and by eventually considering, for part of the network, multiplexing techniques which provide apparent direct links between nonadjacent centers.

This network, which was created mainly to accommodate Type-*B* traffic, is able to forward Type-*A* traffic within the time constraints imposed and, with the proposed studies mentioned, it should be able to cope with the expected increase in this reservations data traffic. Any other solution, such as the building of a dedicated network for Type-A traffic, would have entailed much larger investments.

REFERENCES

[1] W. B. Boehm and R. L. Mobley, "Adaptive Routing Techniques for a Distributed-Communications System," *IEEE Trans. Comm. Tech.* (June 1969).

[2] D. J. Silk, "Principles of Adaptive Routing," Queen's College, Cambridge (1968).

[3] I. Cohen, "Control of Data Processor Networks," *Proc. IEEE ICC Conf.* (June 1970).

[4] H. Frank and W. Chou, "Cost and Throughput Considerations in Computer Communication Networks," *Proc. AFIPS Spring Joint Conf.* (1971).

[5] "Resource Sharing Computer Networks," *Proc. AFIPS Spring Joint Conf.* (May 1970).

[6] G. J. Brandt, "Das Preemptive Warteverlustsystem," Ph.D. Thesis, Universitat Stuttgart (1970).